УДК 621.382.3 (045)

О. С. МЕЛЬНИК, А. М. МИКОЛУШКО

РЕПРОГРАМОВАНІ МУЛЬТИПЛЕКСОРНІ НАНОСХЕМИ

Застосування великих інтегральних схем (ВІС) в цифрових мікро– і наноелектронних пристроях дозволяє істотно поліпшити їх експлуатаційні можливості, в першу чергу підвищити надійність і швидкодію, понизити споживану потужність і габаритні розміри. Проте розробка ВІС є тривалим і дорогим процесом, який економічно виправданий тільки при досить великому обсязі випуску. Підвищення спеціалізації інтегральних схем завжди вступає в протиріччя з їх універсальністю. Усунути вказане протиріччя між спеціалізацією і універсальністю можна шляхом розробки ВІС, алгоритми роботи яких можуть бути змінені за бажанням розробника конкретної апаратури, тобто, шляхом створення логічних схем, що настроюються або програмуються. В роботі реалізовані автоматизовані методи репрограмування мультиплексорних наносхем для відтворення логічних функцій декількох аргументів. На сучасних автоматизованих системах здійснено проектування та верифікацію нанопристроїв.

Ключові слова: мультиплексорні наносхеми, автоматизоване проектування, програмування логічних функцій.

А. С. МЕЛЬНИК, А. М. МИКОЛУШКО

РЕПРОГРАММИРОВАНЫЕ МУЛЬТИПЛЕКСОРНЫЕ НАНОСХЕМЫ

Применение больших интегральных схем (БИС) в цифровых микро– и наноэлектронных устройствах позволяет существенно улучшить их эксплуатационные возможности, в первую очередь повысить надежность и быстродействие, снизить потребляемую мощность и габаритные размеры. Однако разработка БИС является длительным и дорогостоящим процессом, который экономически оправдан только при достаточно большом объеме выпуска. Повышение специализации интегральных схем при улучшении указанных выше показателей всегда вступает в противоречие с их универсальностью. Устранить указанное противоречие между специализацией и универсальностью можно путем разработки БИС, алгоритмы, работы которых могут быть изменены по желанию разработчика конкретной аппаратуры, то есть, путем создания логических схем, которые настраиваются или программируются. В работе реализованы автоматизированные методы репрограммирования мультиплексорных наносхем для воспроизведения логических функций нескольких аргументов. На современных автоматизированных системах осуществлено проектирование и верификация наноустройств.

Ключевые слова: мультиплексорные наносхемы, автоматизированное проектирование, программирование логических функций.

O. S. MELNYK, A. M. MIKOLUSHKO

REPROGRAMMABLE MULTIPLEXER NANOCIRCUITS

The use of large-scale integrated circuits (LSIC) in digital micro- and nanoelectronic devices can significantly improve their operational capabilities, primarily improve reliability and performance, reduce power consumption and overall dimensions. However, the development of the LSIC is a long and costly process, which is economically justified only with a fairly large volume of output of finished products. Increasing the specialization of IC contradicts their universality. Eliminating this contradiction between specialization and universality can be through the development of a LSIC, the algorithms of which can be changed at the request of the developer of specific equipment, that is, by programmed logic circuits. In the paper automated methods of multiplexer nanoscale reprogramming for reproducing logical functions of several arguments are realized. The design and verification of nanodevices are realized on modern automated systems

Key words: multiplexer nanocircuits, computer–aided design, programmable logic function.

Вступ. Під *програмованістю* розуміється не здатність реалізувати заданий алгоритм обробки вхідних кодів, змінюючи програму роботи, як це робить мікропроцесор, а можливість зміни внутрішньої структури *великої інтегральної схеми* (ВІС) так, щоб вона забезпечувала реалізацію заданих логічних функцій на апаратному рівні.

При виготовленні таких ВІС використовується єдиний комплекс фотошаблонів, тому з точки зору виробника це – універсальні вироби. Налаштування ж даної ВІС на заданий алгоритм роботи виконує безпосередньо виробник апаратури, з точки зору якого ця схема реалізує вузько спеціалізовані завдання. В результаті програмування в *інтегральній схемі* (ІС) вносяться зворотні або незворотні зміни структури, які і приводять до набуття заданих характеристик. Відповідно до сказаного, основною перевагою *програмованих логічних інтегральних схем* (ПЛІС) перед спеціалізованими ВІС являється малий час виготовлення ВІС з наперед заданими характеристиками. Залежно від рівня складності одна ПЛІС може замінити до 10 тис. і більше ІС малого і середнього ступенів інтеграції. Отже, така заміна дозволяє значною мірою реалізувати переваги ВІС при низькій вартості виготовлення, що особливо важливо при невеликих обсягах випуску конкретної апаратури.

Метою цієї роботи є розробка методів програмування ПЛІС шляхом надмірності їх апаратної частини, тобто введення додаткових виводів і структур налаштування, додавання інформаційних ланцюгів тощо. Реальна швидкодія пристроїв, виконаних на ПЛІС, їх споживання і інші характеристики будуть завжди гірші, ніж у пристроїв на спеціалізованих ВІС. Проте ці характеристики будуть свідомо кращі за аналогічні характеристики апаратури, побудованої на стандартних ІС.

Репрограмування мультиплексорів в якості логічних елементів. В якості простих ПЛІС можуть використовуватися *мультиплексори*. Суть використання мультиплексора в якості універсального логічного наноелементу полягає в тому, що його адресні входи використовуються як інформаційні і на них подаються аргументи відтворюваної функції, а інформаційні входи виконують роль репрограмованих (настроювальних).

Наприклад, реалізуємо на мультиплексорі (4→1) логічну функцію двох аргументів «*Виключне АБО*» (дода-

вання по модулю 2 чи нерівнозначності):

$$f_{\oplus} = x_1\overline{x}_0 \vee \overline{x}_1 x_0 = x_1 \oplus x_0, \quad (1)$$

задану у вигляді таблиці істинності (табл. 1).

Таблиця 1 – Таблиця істинності операції «Виключне АБО»

x_1	x_0	$f_{\oplus}$
0	0	0
0	1	1
1	0	1
1	1	0

Таблиця 2 – Таблиця істинності мультиплексора (4→1)

x_1	x_0	$f_{\oplus}$
0	0	$D0$
0	1	$D1$
1	0	$D2$
1	1	$D3$

Використовуючи рівняння вихідної функції цього мультиплексора, а саме:

$$f = D3(x_1x_0) \vee D2(x_1\overline{x}_0) \vee D1(\overline{x}_1x_0) \vee D0(\overline{x}_1\overline{x}_0), \quad (2)$$

та його таблицю істинності (табл. 2), порівняємо між собою функції (1), (2) і табл. 1, 2 та запишемо функцію програмування мультиплексора у вигляді:

$$f_{пр} = 0(x_1x_0) \vee 1(x_1\overline{x}_0) \vee 1(\overline{x}_1x_0) \vee 0(\overline{x}_1\overline{x}_0). \quad (3)$$

В результаті реалізуємо цей алгоритм програмування у вигляді табл. 3 і схеми мультиплексора, побудованої на рис. 1. Напруга живлення $U_Ж$ реалізує логічну одиницю, а заземлення – логічний нуль.

Таблиця 3 – Значення логічної функції «Виключне АБО»

x_1	x_0	$f_{пр}$	D	f
0	0	0	$D0 = 0$	0
0	1	1	$D1 = 1$	1
1	0	1	$D2 = 1$	1
1	1	0	$D3 = 0$	0

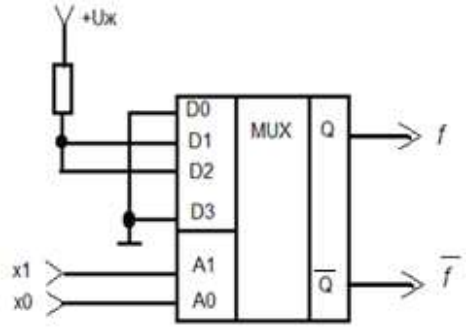


Рис. 1 – Реалізація логічної функції двох вхідних змінних на мультиплексорі.

На мультиплексорі (4→1) можливе репрограмування 16 найпростіших функцій двох аргументів (табл. 4). Для цього на адресні входи A1 та A0 слід подавати бінарні аргументи x_1 та x_0, а інформаційні $D3, ..., D0$ – репрограмувати логічними константами цих аргументів.

З розглянутого прикладу видно, що описаний метод обмежується реалізацією функцій чотирьох змінних, оскільки мультиплексори, що реально випускаються, мають не більше 16 інформаційних входів. Таким чином, для реалізації

$$2^{2^4} = 2^{16} = 65536$$

логічних функцій чотирьох вхідних змінних можна скористатися *мультиплексорною наносхемою*, яка буде універсальним логічним елементом, оскільки без використання додаткових апаратних засобів дозволяє реалізувати логічну функцію довільного виду.

Для порівняння приклад реалізації операції (1) з використанням п'яти елементів НІ, І та АБО наведений на рис. 2.

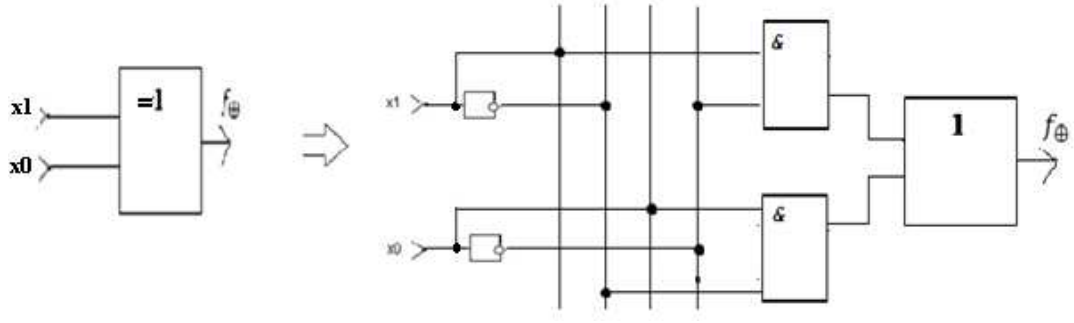


Рис. 2 – Схемна реалізація операції «Виключне АБО».

Таблиця 4 – Функції двох аргументів для репрограмування мультиплексорів

x_1	1	1	0	0	Нормальна диз'юнктивна форма	Логічна функція
x_0	1	0	1	0		
№	*D*	*D*	*D*	*D*		
0	0	0	0	0	$f_0 = 0$	Активний логічний 0
1	0	0	0	1	$f_1 = \overline{x}_1\overline{x}_0 = \overline{x_1 \vee x_0} = x_1 \downarrow x_0$	АБО-НІ (*стрілка Пірса*)
2	0	0	1	0	$f_2 = \overline{x}_1 x_0 = x_0 \leftarrow x_1$	Заборона
3	0	0	1	1	$f_3 = \overline{x}_1 x_0 \vee \overline{x}_1\overline{x}_0 = x_1(x_0 \vee \overline{x}_0) = \overline{x}_1$	Заперечення x_1
4	0	1	0	0	$f_4 = x_1\overline{x}_0 = x_1 \leftarrow x_0$	Зворотна заборона
5	0	1	0	1	$f_5 = x_1\overline{x}_0 \vee \overline{x}_1\overline{x}_0 = x_0(x_1 \vee \overline{x}_1) = \overline{x}_0$	Заперечення x_0
6	0	1	1	0	$f_6 = x_1\overline{x}_0 \vee \overline{x}_1 x_0 = x_1 \oplus x_0$	Виключне АБО (нерівнозначність)
7	0	1	1	1	$f_7 = x_1\overline{x}_0 \vee \overline{x}_1 x_0 \vee \overline{x}_1\overline{x}_0 = x_1\overline{x}_0 \vee \overline{x}_1 = x_1\overline{x}_0 \vee \overline{x}_1(\overline{x}_0 \vee 1) = x_1\overline{x}_0 \vee \overline{x}_1\overline{x}_0 \vee \overline{x_1 x_1} = x_1\overline{x}_0 \vee x_1\overline{x}_1 \vee \overline{x}_1\overline{x}_0 \vee \overline{x_1 x_1} = x_1(\overline{x}_0 \vee \overline{x}_1) \vee \overline{x}_1(x_0 \vee \overline{x}_1) = (x_0 \vee \overline{x}_1)(\overline{x}_1 \vee \overline{x}_0) = \overline{x_1 x_0} = x_1 \mid x_0$	І-НЕ (*штрих Шеффера*)
8	1	0	0	0	$f_8 = x_1 x_0$	І
9	1	0	0	1	$f_9 = x_1 x_0 \vee \overline{x}_1\overline{x}_0 = \overline{x_1 \oplus x_0}$	Виключне АБО-НІ (рівнозначність)
10	1	0	1	0	$f_{10} = x_1 x_0 \vee \overline{x}_1 x_0 = x_0(x_1 \vee \overline{x}_1) = x_0$	Активний аргумент x_0
11	1	0	1	1	$f_{11} = x_1 x_0 \vee \overline{x}_1 x_0 \vee \overline{x}_1\overline{x}_0 = x_0(x_1 \vee \overline{x}_1) \vee \overline{x}_1\overline{x}_0 = \overline{x}_1\overline{x}_0 \vee x_0(\overline{x}_1 \vee 1) = x_1(x_0 \vee \overline{x}_0) \vee \overline{x}_0 x_0 \vee x_0 x_0 = \overline{x}_1(x_0 \vee \overline{x}_0) \vee x_0(x_0 \vee \overline{x}_0) = (x_0 \vee \overline{x}_1)(x_0 \vee \overline{x}_0) = \overline{x}_1 \vee x_0 = x_1 \rightarrow x_0$	Імплікація
12	1	1	0	0	$f_{12} = x_1 x_0 \vee x_1\overline{x}_0 = x_1(x_0 \vee \overline{x}_0) = x_1$	Активний аргумент x_1
13	1	1	0	1	$f_{13} = x_1 x_0 \vee \overline{x}_1 x_0 \vee \overline{x}_1\overline{x}_0 = x_1(x_0 \vee \overline{x}_0) \vee \overline{x}_1\overline{x}_0 = \overline{x}_1\overline{x}_0 \vee x_1(\overline{x}_1 \vee 1) = \overline{x}_1(x_1 \vee \overline{x}_0) \vee x_1(\overline{x}_1 \vee 1) = \overline{x}_1(x_1 \vee \overline{x}_0) \vee \overline{x}_0 x_1 \vee x_1 x_1 = (x_1 \vee \overline{x}_0)(x_1 \vee \overline{x}_1) = x_1 \vee \overline{x}_0 = x_0 \rightarrow x_1$	Зворотна імплікація
14	1	1	1	0	$f_{14} = x_1 x_0 \vee x_1\overline{x}_0 \vee \overline{x}_1 x_0 = (x_1 \vee x_0)(x_1 \vee \overline{x}_1) = x_0 \vee x_1$	АБО
15	1	1	1	1	$f_{15} = x_1 x_0 \vee x_1\overline{x}_0 \vee \overline{x}_1 x_0 \vee \overline{x}_1\overline{x}_0 = (x_1 \oplus x_0) \vee \overline{(x_1 \oplus x_0)} = 1$	Активна логічна 1

При необхідності реалізації логічної функції великого числа вхідних змінних можна скористатися структурою *мультиплексорного дерева.* Проте при невеликому числі аргументів цю задачу можна вирішувати і іншим методом, а саме вибором сигналів налаштування не з множини $\{1, 0\}$, як це було зроблено вище, а з множини $\{1, 0, x_i\}$, де x_i – це один з аргументів відтворюваної функції. В цьому випадку вдається на мультиплексорній наносхемі без додаткових апаратних витрат реалізувати логічну функцію, число аргументів якої на одиницю більше числа його адресних входів.

Наприклад, на мультиплексорі (4→1) реалізуємо логічну функцію трьох вхідних змінних наступного вигляду:

$$f = x_2 \vee x_1\overline{x}_0 \vee \overline{x}_1 x_0 . \qquad (4)$$

Для отримання *диз'юнктивної нормальної форми* (ДНФ) логічних функцій використовують дужкові перетворення, тобто неповні добутки (*терми*) домножують на одиночні суми недостаючих аргументів та функцій: $(x_i \vee \overline{x}_i)$, $(x_i \vee 1)$, $(f_i \vee \overline{f}_i)$, $(f_i \vee 1)$ або додають нульові добутки $(x_i\overline{x}_i)$, $(x_i 0)$, $(f_i\overline{f}_i)$, $(f_i 0)$.

Перетворимо за цими правилами задану функцію (4):

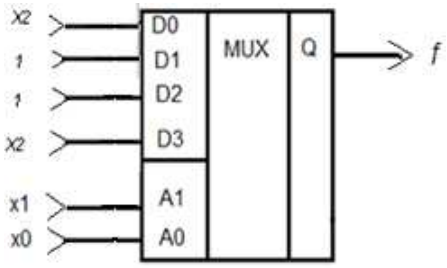
$$f = x_2(x_1 \vee \overline{x}_1)(x_0 \vee \overline{x}_0) \vee x_1\overline{x}_0 \vee \overline{x}_1 x_0 = x_2x_1x_0 \vee x_2x_1\overline{x}_0 \vee x_2\overline{x}_1x_0 \vee x_2\overline{x}_1\overline{x}_0 \vee x_1\overline{x}_0 \vee \overline{x}_1x_0 =$$

$$= x_2x_1x_0 \vee x_1\overline{x}_0(x_2 \vee 1) \vee \overline{x}_1x_0(x_2 \vee 1) \vee x_2\overline{x}_1\overline{x}_0 = x_2(x_1x_0) \vee 1(x_1\overline{x}_0) \vee 1(\overline{x}_1x_0) \vee x_2(\overline{x}_1\overline{x}_0). \quad (5)$$

Отже отримано функцію програмування (5) мультиплексора (4→1), у якого на адресні входи подано сигнали змінних x_1 і x_0. Таблиця і схема програмування у цьому прикладі наведені на рис. 3.

x_1	x_0	D
0	0	$D0 = x_2$
0	1	$D1 = 1$
1	0	$D2 = 1$
1	1	$D3 = x_2$

а



б

Рис. 3 – Функції трьох вхідних змінних: *а* – таблиця програмування; *б* – схемна реалізація.

Слід зауважити, що таке схемотехнічне рішення не є єдиним. На адресні входи A1 та A0 можна подавати довільні аргументи x_2x_0 і x_2x_i, а на програмовані входи $D3, ..., D0$ – залишкові аргументи чи навіть функції.

З точки зору апаратних витрат це схематичне рішення (рис. 3, *б*) є оптимальним.

Приклад моделювання. Реалізуємо на мультиплексорних мікро– та наносхемах (4→1) трьохаргументну логічну функцію мажоритарного вибору двох із трьох аргументів. На виході *мажоритарного елемента* (МЕ) логічний сигнал співпадає з більшістю сигналів на непарній кількості входів [1]. На рис. 4 показане умовне позначення тривходового МЕ, а в табл. 5 побудована відповідна таблиця істинності.

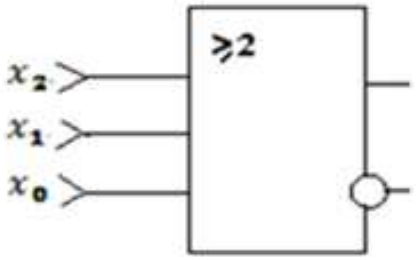


Рис. 4 – Трьохвходовий МЕ.

Таблиця 5 – Таблиця істинності МЕ

x_2	x_1	x_0	f	$\overline{f}$
0	0	0	0	1
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0

Диз'юнктивну нормальну форму (ДНФ) мажоритарної функції f отримують з її таблиці істинності за правилом:

$$f = maj(x_2, x_1, x_0) = \overline{x}_2x_1x_0 \vee x_2\overline{x}_1x_0 \vee x_2x_1\overline{x}_0 \vee x_2x_1x_0. \quad (6)$$

Виконавши дужкові перетворення рівняння (6) шляхом триразового логічного додавання останнього доданку, знаходять *довершену форму мажоритарної функції* наступного типу:

$$f = maj(x_2, x_1, x_0) = x_2x_1(\overline{x}_0 \vee x_0) \vee x_2x_0(\overline{x}_1 \vee x_1) \vee x_1x_0(x_2 \vee x_2) = x_2x_1 \vee x_2x_0 \vee x_1x_0. \quad (7)$$

Мажоритарна функція мінімальної форми (7) є найбільш поширеною в наноелектроніці. Інверсну мажоритарну функцію називають *міноритарною*:

$$f = maj(\overline{x_2, x_1, x_0}) = min(x_2, x_1, x_0) = \overline{x}_2\overline{x}_1 \vee \overline{x}_2\overline{x}_0 \vee \overline{x}_1\overline{x}_0. \quad (8)$$

Якщо один з входів МЕ запрограмувати сигналом $x_2 = 0$ чи $x_2 = 1$, то як видно з табл. 5, на прямому виході МЕ реалізуються логічні функції помноження І ($f_1 = x_1x_0$) чи додавання АБО ($f_{АБО} = x_1 \vee x_0$).

Перетворимо повну мажоритарну функцію (6) для програмування мультиплексора (4→1) при використанні в якості адресних аргументів x_1 і x_0. Виконаємо спрощені дужкові перетворення цієї функції:

$$f = maj(x_2, x_1, x_0) = x_1x_0(\overline{x}_2 \vee x_2) \vee x_2x_1\overline{x}_0 \vee x_2\overline{x}_1x_0 = x_1x_0 \vee x_2x_1\overline{x}_0 \vee x_2\overline{x}_1x_0. \quad (9)$$

Порівнюючи вихідну функцію мультиплексора (2) з перетвореною МЕ (9), запишемо функцію програмування мультиплексора (4→1) для реалізації мажоритарної функції:

$$f_{пр} = 1(x_1x_0) \vee x_2(x_1\overline{x}_0) \vee x_2(\overline{x}_1x_0) \vee 0(\overline{x}_1\overline{x}_0). \quad (10)$$

Отже, згідно рівнянню (10) на інформаційні входи мультиплексора слід подавати наступні сигнали $D3 = 1$,

$D2 = D1 = x_2$ та $D0 = 0$.

Рішення (10) не є єдино існуючим. На адресні входи мультиплексора A_1 та A_0 можливо подавати довільні набори аргументів, наприклад x_2x_0 або x_2x_1, а на інформаційні входи $D3 \ldots D0$ – залишкові аргументи та логічні сигнали з множини $\{1, 0\}$. В першому випадку функція програмування матиме вигляд:

$$f_{np}(x_2x_0) = 1(x_2x_0) \vee x_1(\overline{x}_2\overline{x}_0) \vee x_1(\overline{x}_2x_0) \vee 0(\overline{x}_2\overline{x}_0),$$

а в другому:

$$f_{np}(x_2x_1) = 1(x_2x_1) \vee x_0(x_2\overline{x}_1) \vee x_0(x_2\overline{x}_1) \vee 0(\overline{x}_2\overline{x}_1).$$

Таким чином, практичне програмування мажоритарної функції (7) на базі мультиплексорів (4→1) свідчить, що при будь-якій комбінації адресних аргументів на виході реалізується однотипні функції програмування.

Результати комп'ютерного моделювання. На рис. 5 побудована функціональна схема запрограмованого мультиплексора в графічному редакторі системи автоматизованого проектування (САПР) MAX+PLUS II, яка є інтегрованим середовищем для розробки цифрових пристроїв на базі програмованих логічних схем (ПЛІС) фірми ALTERA [2]. Вона складається з восьми мікросхем: трьох інверторів НЕ, чотирьох тривходових логічних елементів І і одного чотиривходового логічного елемента АБО, має два адресних входи, чотири інформаційних і один комплементарний вихід.

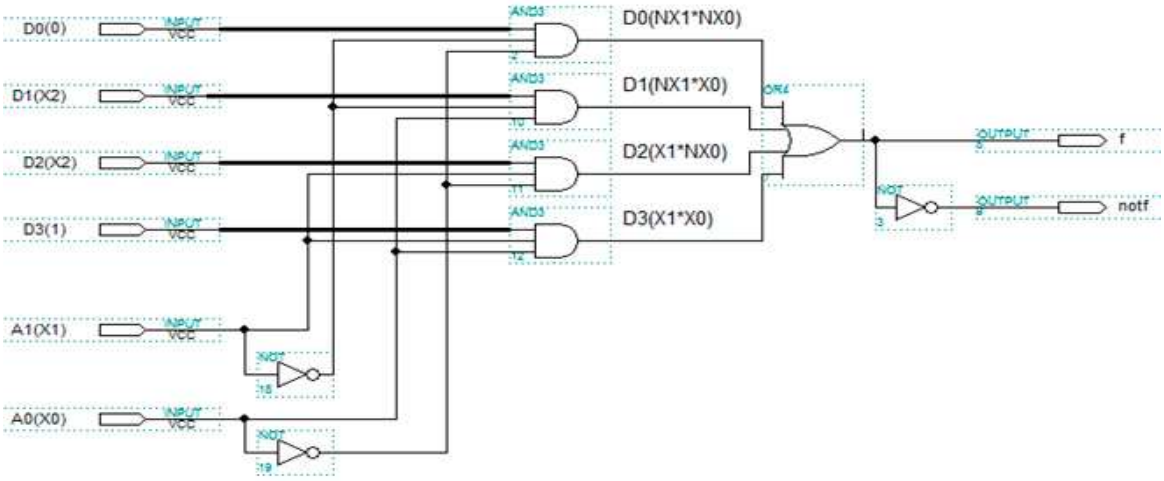


Рис. 5 – Мікросхема запрограмованого мультиплексора.

Часовий аналіз вхідних і вихідних сигналів результатів програмування мажоритарної функції (7) наведений у вікні редактора діаграм на рис. 6. Він повністю відповідає таблиці істинності МЕ (табл. 5).

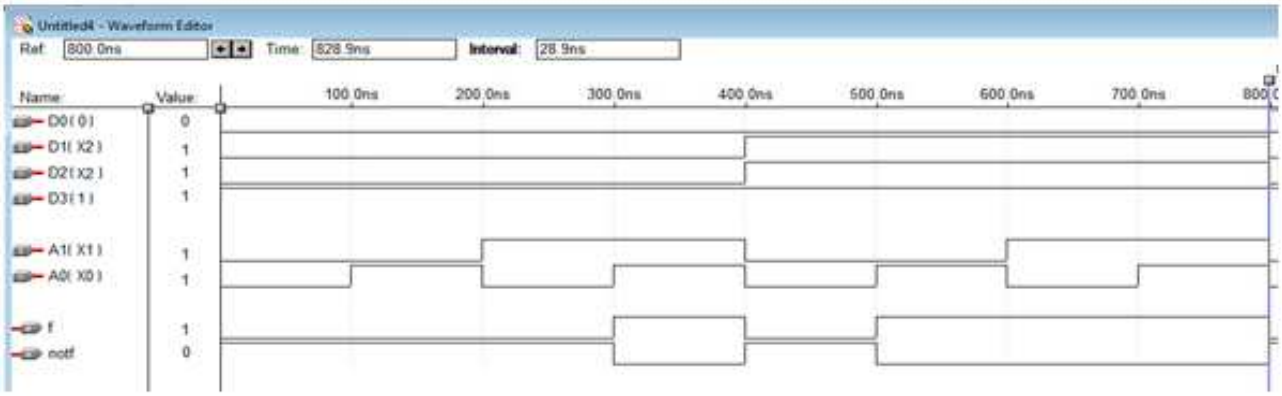
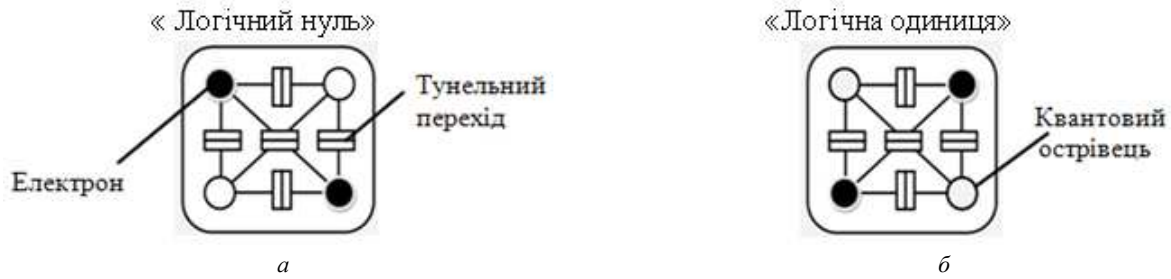


Рис. 6 – Часові діаграми запрограмованого мультиплексора.

Далі синтезуємо одноелектронну наносхему мультиплексора (4→1) на *квантових коміркових автоматах* (КА). Базова діелектрична комірка КА розміром (50×50) нм містить чотири напівпровідникові чи металеві квантові точки (*острівці*), які геометрично розміщені по куткам квадрату, як показано на рис. 7. Комірки використовуються для побудови нанопровідників, мажоритарних елементів, інверторів, запам'ятовуючих пристроїв тощо [3].

Принцип дії КА, заснований на квантовому тунелюванні та кулонівській взаємодії електронів, дозволяє створювати пристрої, які можуть працювати як комутуючі транзистори, але з меншими розмірами та меншою споживаною потужністю. Однорідна архітектура КА може бути сформована за допомогою електронної літографії.

Кожна комірка може містити два надлишкові електрони, які в змозі тунелювати між квантовими точками. Коли потенціал діелектричних перетинок між точками високий, електрони будуть локалізовані та не зможуть тунелювати, а коли потенціал низький, то електрони зможуть вільно тунелювати. Кулонівське відштовхування між електронами примусить їх розташуватися на діаметрально протилежних кутках з двома можливими конфігураціями (рис. 7). Ці два стабільні стани можуть бути представлені як поляризації комірок $P=-1$ та $P=+1$. Рівень поляризації $P=-1$ відповідає логічному нулю (рис. 7, *а*), а $P=+1$ відповідає логічній одиниці (рис. 7, *б*). Якщо в масиві КА, отриманому після розміщення декількох комірок в ряд, змінити поляризацію першої комірки в визначений стан, то друга комірка прийме той же стан по принципу найменшої енергії. Далі по принципу доміно всі комірки масиву набудуть цей же рівень поляризації.



а *б*

Рис. 7 – Базова комірка в двох можливих логічних станах: а – логічний нуль; б – логічна одиниця.

Поляризація комірки P визначається співвідношеннями:

$$P=\frac{\rho_1-\rho_2+\rho_3-\rho_4}{\rho_1+\rho_2+\rho_3+\rho_4}=\frac{(\rho_1+\rho_3)-(\rho_2+\rho_4)}{(\rho_1+\rho_3)+(\rho_2+\rho_4)},$$

де ρ_1 і ρ_3, ρ_2 і ρ_4 – щільності зарядів електронів у діагонально розташованих квантових точках комірки.

Таким чином, бінарна інформація передається без руху зарядів. Тобто нема протікання струму між комірками. Це – основна причина, чому структури КА споживають надмізерну кількість енергії $\sim 10^{-22}$ Дж.

На рис. 8 наведена наносхема мультиплексора на 3 інверторах і 11 МЕ, яка запрограмована для реалізації мажоритарної функції (7).

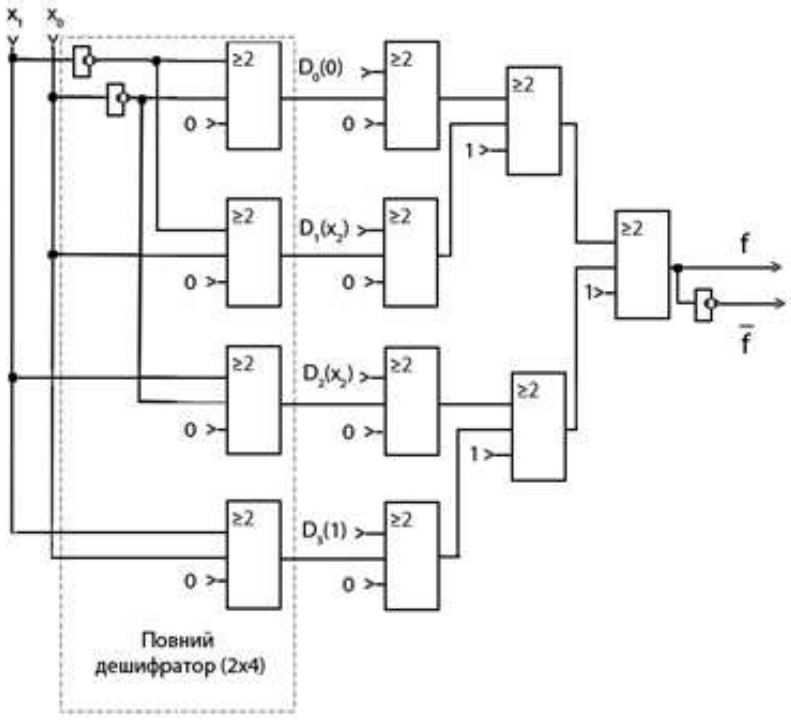
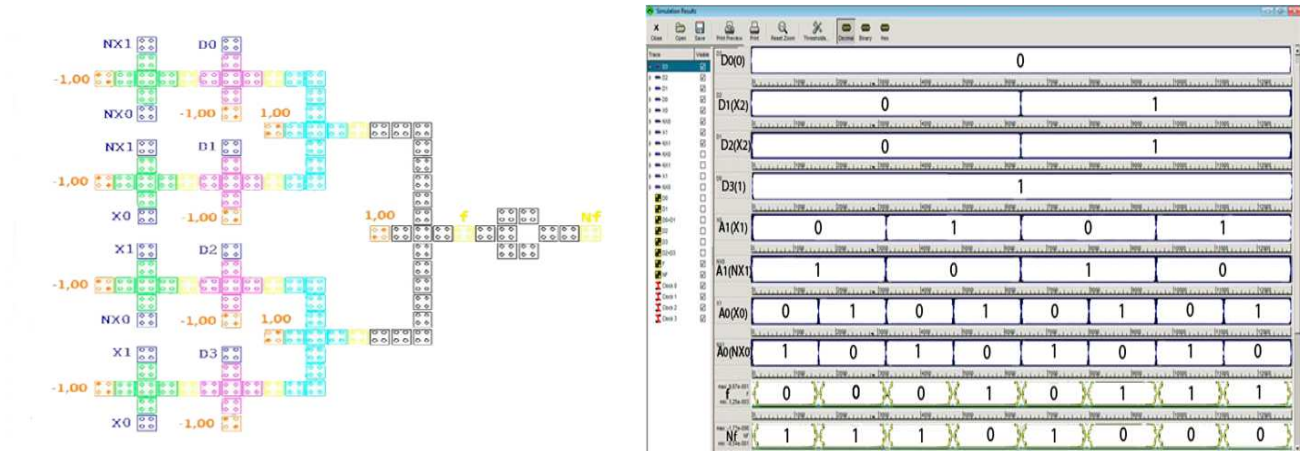


Рис. 8 – Мажоритарна наносхема запрограмованого мультиплексора.

На рис. 9, *а* побудована мажоритарна наносхема на КА запрограмованого мультиплексора (4→1), яка реалізує функцію мажоритарного вибору, а на рис. 9, *б* – часові діаграми, отримані в результаті моделювання на САПР QCA Designer [3]. Сигнали вихідної функції f повністю співпадають з табличними значеннями для МЕ (табл. 5).

Запрограмована таким чином наносхема мультипексора (4 – 1) базується на 114 квантових коміркових автоматах, розмір яких складає (18×18) нм, відстань між їх центрами дорівнює 20 нм, з чотирма квантовими острівцями діаметром по 5 нм. Загальна площа наносхеми мультиплексора дорівнює

$$(454{,}8\times362{,}81)\ \text{нм}^2 = 0{,}16\ \text{мкм}^2.$$



а *б*

Рис. 9 – Наносхема мультиплексора: *а* – на квантових коміркових автоматах;
б – часові характеристики.

При розробці складних логічних пристроїв доводиться стикатися з виконанням однотипних операцій І та АБО над різною кількістю змінних. В якості прикладів запрограмуємо мультиплексори (4→1) на виконання логічних функцій помноження та додавання декількох аргументів:

$$f_I = x_n \ldots x_2 x_1 x_0 = \Lambda_{i=0}^{n} x_i\,; \tag{11}$$

$$f_{АБО} = x_n \vee \ldots \vee x_2 \vee x_1 \vee x_0 = \mathrm{V}_{i=0}^{n} x_i\,. \tag{12}$$

Після дужкових перетворень функції (11) та після домножень суми аргументів $[x_n \vee \ldots \vee x_2]$ на одиночні суми недостаючих аргументів $(x_1 \vee \overline{x}_1)$, $(x_0 \vee \overline{x}_0)$, а аргументів x_1 – на $(x_0 \vee \overline{x}_0)$ та x_0 – на $x_1 \vee \overline{x}_1$, відповідно, функції (12), отримують наступні функції програмувань:

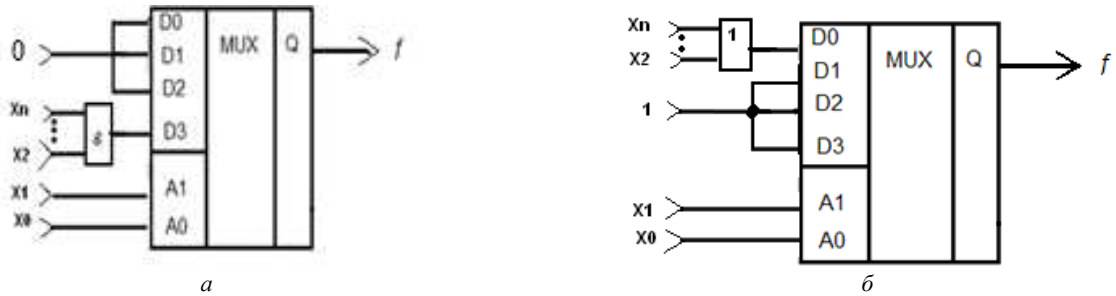
$$f_{пр.I} = [\Lambda_{i=2}^{n} x_i](x_1 x_0) \vee 0(x_1 \overline{x}_0) \vee 0(\overline{x}_1 x_0) \vee 0(\overline{x}_1 \overline{x}_0)\,, \tag{13}$$

$$f_{пр.АБО} = \left[\mathop{\vee}_{i=2}^{n} x_i\right](x_1 \vee \overline{x}_1)(x_0 \vee \overline{x}_0) \vee x_1(x_0 \vee \overline{x}_0) \vee x_0(x_1 \vee \overline{x}_1) =$$

$$= \left[\mathop{\vee}_{i=2}^{n} x_i\right](x_1 x_0 \vee x_1 \overline{x}_0 \vee \overline{x}_1 x_0 \vee \overline{x}_1 \overline{x}_0) \vee (x_1 x_0 \vee x_1 \overline{x}_0 \vee x_1 x_0 \vee \overline{x}_1 x_0) =$$

$$= \left(\left[\mathop{\vee}_{i=2}^{n} x_i\right] \vee 1 \vee 1\right) x_1 x_0 \vee \left(\left[\mathop{\vee}_{i=2}^{n} x_i\right] \vee 1\right) \overline{x}_1 x_0 \vee \left(\left[\mathop{\vee}_{i=2}^{n} x_i\right] \vee 1\right) \overline{x}_1 \overline{x}_0 = 1(x_1 x_0) \vee 1(x_1 \overline{x}_0) \vee 1(\overline{x}_1 x_0) \vee \left[\mathop{\vee}_{i=2}^{n} x_i\right] \overline{x}_1 \overline{x}_0. \tag{14}$$

На рис. 10 наведено схемотехнічні реалізації задач програмування функцій (11) та (12) на мультиплексорах (4→1).



а *б*

Рис. 10 – Схемні реалізації програмування багатоаргументних операцій: *а* – операція І; *б* – операція АБО.

Далі, з використанням отриманих результатів, реалізуємо на мультиплексорі (4→1) складнішу логічну функцію чотирьох аргументів:

$$f = x_3 x_1 \overline{x}_0 \vee x_2 \overline{x}_1 x_0 \vee x_3 x_0 . \quad (15)$$

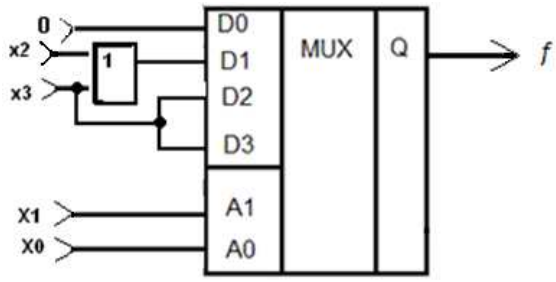
Оберемо в якості адресних змінних x_1 та x_0, і тому домножимо останній неповний добуток (терм) $x_3 x_0$ на одиночну суму недостаючої змінної $(x_1 \vee \overline{x}_1)$:

$$f = x_3 x_1 \overline{x}_0 \vee x_2 \overline{x}_1 x_0 \vee x_2 x_0 (x_1 \vee \overline{x}_1) = x_2 (x_1 x_0) \vee x_2 (x_1 \overline{x}_0) \vee (x_3 \vee x_2)(\overline{x}_1 x_0) \vee 0(\overline{x}_1 \overline{x}_0) . \quad (16)$$

За отриманою функцією (16) на рис. 11 побудовані таблиця та схема програмування мультиплексора, що реалізує початкову функцію (15).

x_2	x_1	D
0	0	$D0 = 0$
0	1	$D1 = x_3 \vee x_2$
1	0	$D2 = x_3$
1	1	$D3 = x_3$

а



б

Рис. 11 – Чотириаргументна функція: *а* – таблиця програмування; *б* – схемотехнічна реалізація.

Використовуючи описаний метод, можна вибирати сигнали репрограмування з ширшої множини, що включає декілька аргументів. При цьому на мультиплексорі з двома адресними входами можна реалізувати логічну функцію трьох, чотирьох і більше змінних. Ефективність використання такого технічного рішення зі збільшенням числа вхідних змінних падає.

Висновки. В роботі розроблені мікро- та наносхеми репрограмованих мультиплексорів для реалізації логічних функцій. Програмування мультиплексора виконувалось за допомогою САПР MAX+PLUS II та САПР QCA Designer. Часові діаграми запрограмованих мультиплексорів в різних середовищах співпали з таблицею істинності.

Принципово нова особливість наноелектроніки пов'язана з тим, що для елементів таких малих розмірів починають переважати квантові ефекти. Відомо, що при переході від мікро- до наноелектроніки квантові ефекти є паразитними, наприклад, роботі класичного транзистора при зменшенні розмірів починає заважати тунелювання носіїв заряду. Проте електроніка, яка використовує квантові ефекти, – це основа нової, так званої наногетероструктурної електроніки.

Список літератури

1. *Мельник О. С.*, *Тодавчич С. В.* Синтез програмованих наноелектронних пристроїв // Електроніка і системи контролю. – 2013. – № 35. – С. 89 – 94.
2. *Стешенко В. Б.* ПЛИС фирмы ALTERA : элементная база, система проектирования и языки описания аппаратуры. – М. : Изд. дом. «Додека – XXI», 2007. – 576 с.
3. Walus K. QCA Designer : A Rapid Design and Simulation Tool for QCA // Int. Journ. of Nanotechn. and Appl. – 2005. – Vol. 2. – № 1. – P. 1 – 7.

References (transliterated)

1. Mel'nyk O. S, Todavchich S. V. Syntez programovanykh nanoelektronnykh prystroyiv [Synthesis of programmable nanoelectronic devices]. *Elektronika i systemy controlyu* [Electronics and Control Systems]. 2013, no. 35, pp. 89–94.
2. Steshenko V. B. *PLIS firmy ALTERA : elementnaya baza, sistemy proektirovaniya i yazyki opisaniya apparatury* [FPGA by ALTERA firm: element base, design system and, hardware description languages]. Moscow, Izd. dome. "Dodeca – XXI" Publ., 2007. 576 p.
3. Walus K. QCA Designer : A Rapid Design and Simulation Tool for QCA. *Int. Journ. of Nanotechn. and Appl*. 2005, vol. 2, no. 1, pp. 1–7.

Надійшла (received) 03.02.2019

Відомості про авторів / Сведения об авторах / Information about authors

Мельник Олександр Степанович (Мельник Александр Степанович, Melnyk Oleksandr Stepanovych) – кандидат технічних наук, доцент, Національний авіаційний університет, м. Київ; тел.: (067) 213-03-08; e-mail: melnyk.ols@gmail.com.

Миколушко Андрій Миколайович (Миколушко Андрей Николаевич, Mykolushko Andriy Mykolayovich) – асистент, Національний авіаційний університет, м. Київ; тел.: (063) 784-26-34; e-mail: 9shik@nau.edu.ua.